

# LAB#6

---

Search

---

# Sequential Search

---

## Sequential Search :

- **Sequential Search** : Search an array or list by checking items one at a time.
  - Linear search is usually very simple to implement, and is practical when the list has only a few elements, or when performing a single search in an unordered list.
-

# Sequential Search

---

```
#include <iostream>
using namespace std;
int LinearSearch(int Array[],int Size,int ValToSearch)
{
    bool NotFound = true;
    int i = 0;
    while(i < Size && NotFound)
    {
        if(ValToSearch != Array[i])
            i++;
        else NotFound = false;    }
    if( NotFound == false )
        return i;
    else return -1;}

```

---

# Sequential Search

```
int main(){
    int Number[] = { 67, 278, 463, 2, 4683, 812, 236, 38 };
    int Quantity = 8;
    int NumberToSearch = 0;
    cout << "Enter the number to search: ";
    cin >> NumberToSearch;
    int i = LinearSearch(Number, Quantity, NumberToSearch);
    if(i == -1) cout << NumberToSearch << " was not found in the
collection\n\n";
    else { cout << NumberToSearch << " is at the " << i+1;
    if( i == 0 ) cout<< "st position of the collection\n\n";
    else if( i == 1 ) cout<< "nd position of the collection\n\n";
    else if( i == 2 ) cout<< "rd position of the collection\n\n";
    else cout<< "th position of the collection\n\n"; }
    return 0; }
```

# Binary Search

---

## Binary Search :

- **Binary Search** >>> sorted array.
  - **Binary Search Algorithm :**
    - 1.get the middle element.
    - 2.If the middle element equals to the searched value, the algorithm stops.
    - 3.Otherwise, two cases are possible:
      - searched value is less, than the middle element. Go to the step 1 for the part of the array, before middle element.
      - searched value is greater, than the middle element. Go to the step 1 for the part of the array, after middle element.
-

# Binary Search

---

**Example Binary Search :**

**12**

1	5	7	12	15	20	25	27	35	40	47	60
0	1	2	3	4	5	6	7	8	9	10	11

1	5	7	12	15
0	1	2	3	4

12	15
3	4

12
----

3

---

# Binary Search

```
#include <iostream>
using namespace std;

int binarySearch(int arr[], int value, int left, int right)
{
    while (left <= right)
    {
        int middle = (left + right) / 2; // compute mid point.
        if (arr[middle] == value) // found it. return position
            return middle;
        else if (arr[middle] > value) // repeat search in bottom half.
            right = middle - 1;
        else
            left = middle + 1; // repeat search in top half.
    }
    return -1;
}
```

# Binary Search

---

```
void main()
{
    int x=0;
    int myarray[10]={2,5,8,10,20,22,26,80,123,131};
    cout<<"Enter a searched value : ";
    cin>>x;
    if(binarySearch(myarray,x,0,9)!=-1)
        cout<<"The searched value found at position : 
        "<<binarySearch(myarray,x,0,9)<<endl;
    else
        cout<<"Not found"<<endl;
}
```

---